**IN THE CLAIMS:**

The following listing of claims will replace all prior versions, and listings, of claims in the application.

1. (currently amended): A graphics system comprising:

a set of industry standard graphics [[accelerators]] cards, wherein each card comprises a rendering processor, an internal frame buffer, and a video data port; and

a series of filtering units, wherein each of the filtering units [[graphics accelerators]] couples to a video data port of a corresponding one of the [[filtering units;]] graphics cards, and wherein each of the graphics [[accelerators]] cards is configured to:

(a) generate a stream of samples in response to received graphics primitives,

(b) add a corresponding dither value to the color components of the samples to obtain dithered color components,

(c) buffer the dithered color components in [[an]] the internal frame buffer, and

(d) forward truncated versions of the dithered color components to the corresponding filtering unit; and

wherein the filtering units are configured to perform a weighted averaging computation on the truncated dithered color components to determine pixel color components.

2. (original): The graphics system of claim 1, wherein each of the graphics accelerators receives the same set of graphics primitives.

3. (original): The graphics system of claim 1, wherein the dither values corresponding to the set of graphics accelerators have an average value of ½.

4. (original): The graphics system of claim 1, wherein the dither values corresponding to the set of graphics accelerators have an average value of 2 to a power J, wherein J is an integer.

5. (original): The graphics system of claim 1, wherein the dither values corresponding to the set of graphics accelerators have a dither radius greater than or equal to one.

6. (original): The graphics system of claim 1, wherein each of the filtering units are configured to support the weighted averaging computation by computing partial sums corresponding to a subset of the samples falling in a filter support region.

7. (original): The graphics system of claim 6, wherein the filtering units are configured to add the partial sums in a pipelined fashion.

8. (original): The graphics system of claim 7, wherein a last of the filtering units in said series is configured to normalize a set of final cumulative sums resulting from said addition of the partial sums in a pipelined fashion.

9. (currently amended): The graphics system of claim 1, wherein [[the set of]] each graphics [[accelerators are industry standard graphics accelerators]] card comprises a plurality of sets of components, wherein each set comprises a rendering processor, an internal frame buffer, and a video data port, and wherein each video data port on the card couples to a corresponding filtering unit.

10. (currently amended): A graphics system comprising:

a set of rendering processors, wherein each rendering processor is connected to a video data output port; and

a series of filtering units, wherein each of the [[rendering processors]] filtering units couples to a corresponding one of the [[filtering units]] video data output ports;

wherein each rendering processor RP(K) of the set of rendering processors is configured to:

(a) generate a stream of samples in response to received graphics primitives,

(b) add a dither value $D_K$ to a data component of each the samples in the stream to obtain dithered data components,

(c) buffer the dithered data components in an internal frame buffer, and

(d) forward a truncated version of the dithered data components to the corresponding filtering unit; and

wherein the filtering units are configured to perform a weighted averaging computation on the truncated dithered data components to determine pixel data components.

11. (original): The graphics system of claim 10, wherein the rendering processors reside within original equipment manufacturer (OEM) graphics cards.

12. (currently amended): The graphics system of claim 11, wherein each of the graphics cards contains two of the rendering processors and two video data output ports, and wherein each video data output port is connected to a different one of the rendering processors.

13. (original): The graphics system of claim 10, wherein the sample data component is a color component.

14. (original): The graphics system of claim 10, wherein the sample data component is an alpha component.

15. (original): The graphics system of claim 10, wherein the dither values corresponding to the set of graphics accelerators have an average value of 2 to a power J, wherein J is an integer.

16. (original): The graphics system of claim 10, wherein each of the filtering units is configured to support the weighted averaging computation by computing a partial sum of the data components corresponding to a subset of the samples falling in a filter support region, wherein the filtering units are configured to add the partial sums in a pipelined fashion.

17 (original): The graphics system of claim 16 wherein a last of the filtering units in said series is configured to normalize a set of final cumulative sums resulting from said addition of the partial sums in a pipelined fashion.

18. (currently amended): A method comprising:

broadcasting a stream of graphics primitives to a set of rendering processors;

each rendering processor RP(K) of said set of rendering processors:

(a) generating a stream of samples in response to received graphics primitives,

(b) adding a dither value $D_K$ to a data component of each <u>of</u> the samples in the stream to obtain dithered data components,

(c) buffering the dithered data components in an internal frame buffer, and

(d) forwarding a truncated version of the dithered data components to a <u>video data output port, wherein each video output port is connected to a</u> corresponding filtering unit; <u>and</u>

the filtering units performing a weighted averaging computation in a pipelined fashion on the truncated dithered data components to determine pixel data components.

19. (original): The method of claim 18, wherein the rendering processors reside within a set of original equipment manufacturer (OEM) graphics cards.

20. (currently amended): The method of claim [[18]] <u>19</u>, wherein each of the graphics cards contains one or more of the rendering processors.

21. (original): The method of claim 18, wherein the data component is a color component.

22. (original): The method of claim 18, wherein the data component is an alpha component.

23. (original): The method of claim 18, wherein the dither values corresponding to the set of graphics accelerators have an average value of 2 to a power J, wherein J is an integer.

24. (currently amended): The method of claim 18 [[10]], wherein each of the filtering units is configured to support the weighted averaging computation by computing a partial sum of the data components corresponding to a subset of the samples falling in a filter support region, wherein the filtering units are configured to add the partial sums in a pipelined fashion.

25. (new): A method comprising:

broadcasting a stream of graphics primitives to a plurality of rendering processors;

each rendering processor RP(K) of said plurality of rendering processors:

(a) generating a stream of samples in response to received graphics primitives,

(b) adding a dither value $D_K$ to a data component of each of the samples in the stream to obtain dithered data components,

(c) buffering the dithered data components in an internal frame buffer, and

(d) forwarding a truncated version of the dithered data components to a corresponding filtering unit of a plurality of filtering units; and

performing a weighted averaging computation in the filtering units in a pipelined fashion on the truncated dithered data components to determine pixel data components, wherein each of the filtering units is configured to support the weighted averaging computation by computing a partial sum from the

data components corresponding to a subset of the samples falling in a filter support region, wherein the filtering units are configured to add the partial sums in a pipelined fashion, and wherein a last of the filtering units in said series is configured to normalize a set of final cumulative sums resulting from said addition of the partial sums in a pipelined fashion.